Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 23 (2011) 451 – 456

Procedia Engineering

www.elsevier.com/locate/procedia

Implement and Analysis of a COMPASS/GPS Software Receiver

Xie Weihua[a]*

[a]Beijing global information centre of application and exploration, Num. 22, Beiqing road, Haidian district, Beijing, 100094, China

Abstract

A multi-channel, multi-system and multi-frequency GNSS software receiver was developed with C++ code. The software receiver takes digital intermediate frequency (IF) data as its input. It performs acquisition, tracking and navigation solution on a standard PC with C++ programs instead of hardware. It is capable of acquisition and tracking of COMPASS B1 C code and GPS L1 C/A code. An IF software signal generator was developed to verify functions and performances of the software receiver. This generator can simulate the COMPASS B1(C code) and GPS L1 (C/A code) constellation signals. The outputs of the signal generator were compared with the outputs of the receiver. The results show that there is less error. The correctness of the software receiver was inspected.

Selection and/or peer-review under responsibility of [name organizer]

Keywords: COMPASS; GPS; Software Receiver; Software Signal Generator; Acquisition; Tracking; C++ Code

1. Introduction

Software-defined radio receivers (SDR) is a concept for transceivers in which the signal processing is accomplished via a programmable general-purpose microprocessor or digital signal processor (DSP), as opposed to an application-specific integrated circuit (ASIC). A software receiver differs from a hardware receiver by performing correlations in software running on a general purpose microprocessor. It can afford batch data processing options that are not available in hardware implementations. New frequencies and new pseudo-random number (PRN) codes can be used simply by making software changes. SDR are not only research tools for the development and test of new navigation and positioning algorithms. The flexibility of software architectures enables them to record several pieces of information that are not limited to position and velocity. Correlator and discriminator outputs, frequency and phase lock indicators and several synchronization messages are just a few examples of the parameters that a software receiver makes available to users and researchers. And the software receiver could be reprogrammed to adjust a new navigation system, which provides an added benefit from the use of the software radio architecture.

* Xie Weihua. Tel.: +86-010-66308515; fax: +86-010-66308509.
E-mail address: xieweihua001979@163.com.

doi:10.1016/j.proeng.2011.11.2529

Along with the decrease of the required processing time, the high configurability, high development speed, low cost software receiver is obtaining more people's favors.

In this paper, a complete Intermediate Frequency (IF) Software COMPASS/GPS Receiver was developed using C++ code. It can be used for acquisition, tracking, and calculating position for the COMPASS B1 C code signal and GPS L1 C/A code signal. To test the functionality and performance of the receiver, an IF software signal generator was developed, which can generate digital IF data for COMPASS B1 and GPS L1 signals.

2. Architecture of Software Defined Receiver

The architecture of software defined receiver is shown in Figure 1. It consists of five modules: an antenna, a RF front-end, acquisition module, 'n' receiver channels and position calculation module. The antenna and RF front-end devices are the only hardware devices of the system. The RF front-end device is necessary to down convert the COMPASS/GPS signal to an intermediate frequency (IF), sample the IF signal and digitize it [1]. The present CPU capacity is still unable to process the COMPASS/GPS signal directly from the antenna in completely software-based approach. Thus a RF front-end device is still necessary. In conventional hardware-based receiver, the three blocks in the dashed textbox in Figure 1 are implemented in an IC chip and hence the user does not have a free access to the algorithms built inside the chips. In software-based receiver, these blocks are fully implemented using high level programming languages and hence the user has complete control over the algorithms. This is the main difference between the software receiver and a conventional hardware receiver. The acquisition module mainly complete three tasks: finding satellites visible to the receiver, finding coarse values for B1 code or L1 C/A code phase and carrier frequency for each satellite [2]. A receiver channel includes six functional blocks: code tracking, carrier tracking, bit synchronization, navigation data decoding, satellite position calculating and pseudo-range calculating, as shown in Figure 2. The detail description of each functional block can be found in references [2].

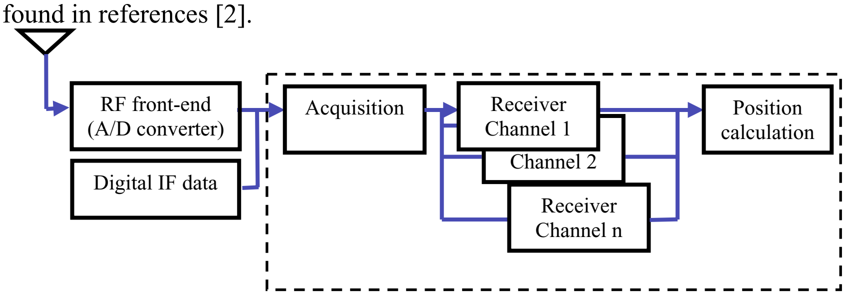

Fig. 1. Architecture of software defined receiver

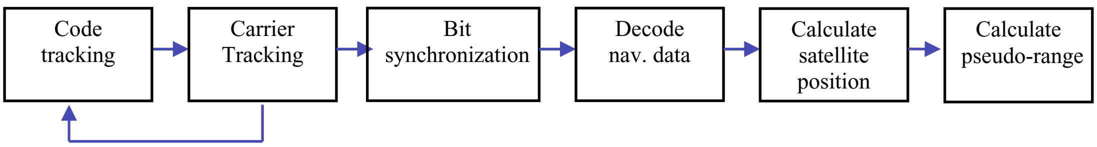

Fig. 2. Functional blocks of a receiver channel

3. IF Software Signal Generator

Due to the satellite launching and networking stage of the COMPASS system，the satellite number in space is few and signals in space can not keep stabilities all the time. To enable the software receiver's functionality and performance test, an IF software signal generator was developed in this article.

The software signal generator consists of a GUI program and a configurable file that gives the user the ability to configure the generated signals. The user decides the number of satellite signals that should be generated. Each signal can be set to be either a COMPASS B1 signal or a GPS L1 signal. The signal can also be described with signal ID, approximate C/N0, carrier Doppler, and delays between different signals. The data of the signal can also be specified. The front end can be described by sampling frequency, intermediate frequency, analog bandwidth, and the number of bits per sample. The generator can also produce signals that contain enough navigation data to give a valid position solution when used in a software receiver. This is done by using COMPASS or GPS ephemeris to describe the satellites' positions and specifying a time and a user position in the signal generator. The delays and Doppler of the different signals are then used to produce the valid position signal. A running interface of generating B1 signal is shown in the Figure 3.

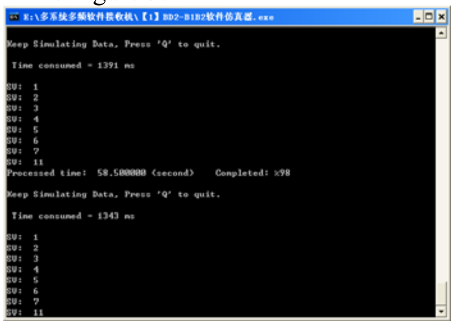

Fig. 3. Running interface of generating B1 signal satellite

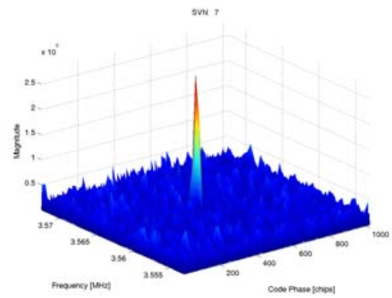

Fig. 4. Result of acquisition for COMPASS SVN 7

4. Implement of Software Receiver

4.1. Implement of acquisition module

The COMPASS signal is combination of carrier wave, C code and navigation message. In order to extract the navigation message from the COMPASS signal, it is necessary to remove the carrier wave (demodulate) and the C code (de-spread). After, demodulation and de-spreading, we get only the navigation message and noise. These navigation messages are then interpreted based on the definitions or coding scheme provided in the interface control document (ICD) of COMPASS. One common way to start an acquisition program is to search for satellites that are visible to the receiver [3]. If the rough location and the approximate time of day are known, information is available on which satellites are available, or can be computed from a recently recorded almanac broadcast. If one uses this method for acquisition, only a few satellites need to be searched.

The conventional approach to perform signal acquisition is through hardware in the time domain. The acquisition is performed on the input data in a continuous manner. Once the signal is found, the information will immediately pass to the tracking hardware. In most receivers the acquisition can be performed on many satellites in parallel. In the software receiver developed in this article, the acquisition is performed on a block of data by FFT method in post-processing mode. The acquisition module contains look-up table which contains FFT processed carrier and PRN code, and FFT implementation related tools. The FFT results of carrier and C code are stored in look-up table with 1ms unit to save calculation time. The details of the algorithm can be seen in article [4]. When the desired signal is found, the information is passed on to the tracking program. Figure 4 shows the result of acquisition for SVN 7 satellite simulated by IF software signal generator.

4.2. Implement of tracking module

The tracking loop follows the incoming signal and adjusts itself to de-spread and de-modulate the incoming signal. If the receiver is stationary, the rate of change of Doppler frequency is small and hence the update rate of tracking loop is also small. In order to track the incoming COMPASS signal, we need to use two tracking loops. Delay Lock Loop (DLL) is used to track the B1 C or L1 C/A code (de-spread) and Phase Lock Loop (PLL) is used to track the frequency of the incoming signal that is related with Doppler frequency. The details of a complete tracking channel on the software receiver can be seen in reference [2][5]. The flow diagram of tracking module implemented in this software receiver is showed in figure 5. And the tracking test results of channel 7 are presented in Figure 6 while the software receiver is running. From the top part of figure 6, we can see that the I_arm can output "0" or "1" navigation data after entering the stable tracking status; and Q_arm outputs noise. The bottom part of figure 6 shows the correlation integration results of early-prompt-late code; the correlation integration result of prompt code is the maximum; the correlation integration results of early and late codes are smaller.

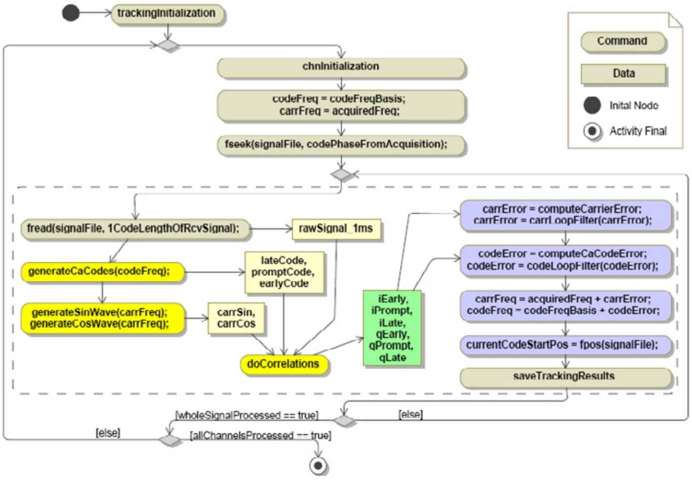

Fig. 5. The flow diagram of tracking module implemented in the software receiver

Fig. 6. The tracking results of channel 7 in the software receiver

4.3. Implement of position computing module

The position computing module mainly includes the following five blocks: bit synchronization, navigation messages decoding, satellite positions calculating, pseudo-ranges calculating, and receiver position calculating. The details of every block's function can be seen in reference [6]. The flow diagram of position computing module implemented in the software receiver is showed in figure 7. The figure 8 shows the position comparison between simulator outputs and software receiver outputs. The figure 9 shows the measurement comparison between simulator outputs and software receiver outputs. From the figure 8 and figure 9, we can see that there is less error between simulator outputs and receiver outputs.

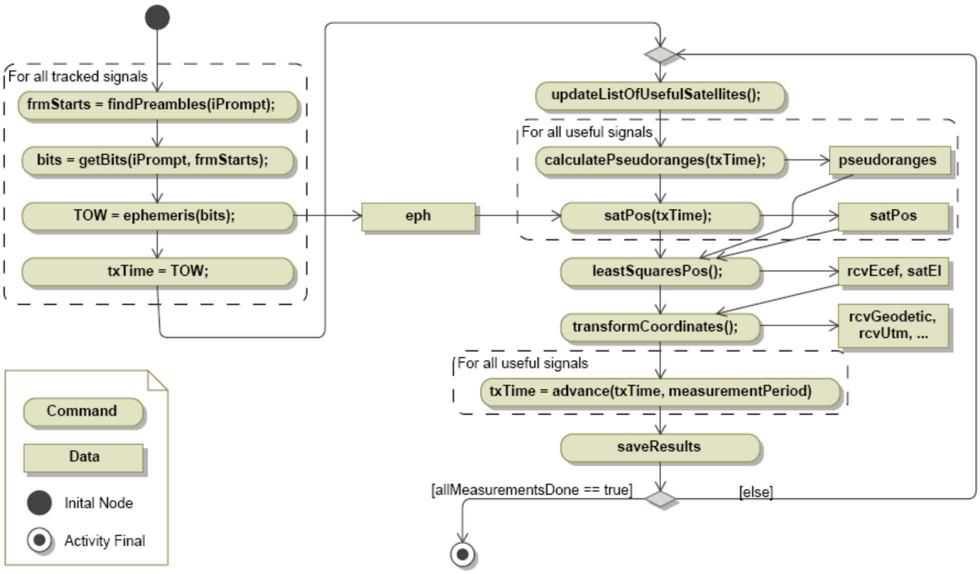

Fig. 7. The flow diagram of position computing module in the software receiver

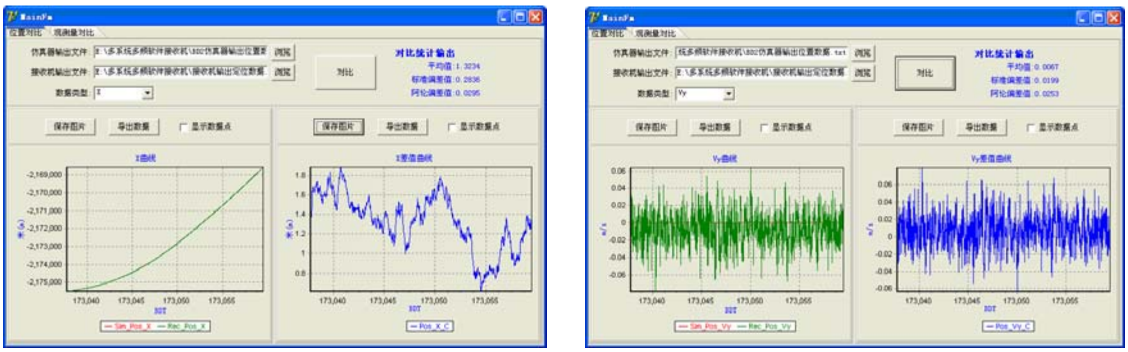

Fig.8. (a) Position comparison between simulator outputs and software receiver outputs; (b) velocity comparison between simulator outputs and software receiver outputs

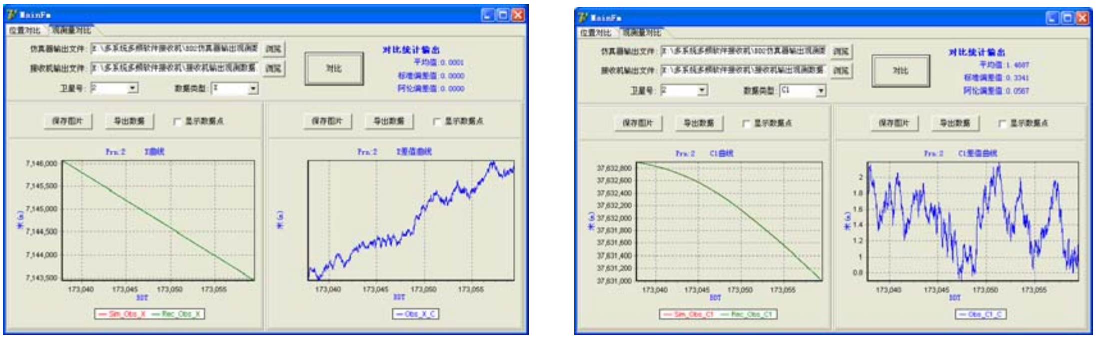

Fig. 9. Measurement comparison between simulator outputs and software receiver outputs

5. Conclusions

Software GNSS receivers offer significant advantages over traditional hardware designs due to their inherent architectural flexibility. A software-defined hybrid COMPASS/GPS receiver, running on a standard PC, is developed and tested in this paper. The receiver is capable of acquisition and tracking of COMPASS B1 C code and GPS L1 C/A code signals. In conjunction with the software receiver, an IF software signal generator was also presented. This generator can simulate the COMPASS B1(C code) and GPS L1 (C/A code) constellation signals, and is used to verify receiver performance. The output results show that software receiver is a feasible choice for hybrid COMPASS/GPS applications. GNSS software receivers are an emerging technology enabling the creation of new applications. The continued demand for more information and modernized radio navigation systems, combined with the exponential evolution of processing power may well see the extinction of some application specific integrated circuits.

Acknowledgement

The author wishes to thank the China Postdoctoral Science Foundation’s fund to this article.

References

[1] James Bao-Yen Tsui. *Fundamentals of Global Positioning System Receivers-A Software Approach*. 2nd ed. New York: John Wiley & Sons, Inc; 2005.

[2] Kai Borre, Dennis M. Akos, Nicolaj Bertelsen, Peter Rinder, Soren Holdt Jensen. *A Software-Defined GPS and Galileo Receiver*. Birkhauser Boston; 2007.

[3] ZhangWei, ZhangKe, WuBinbin, Suh Heejong. Simulation and Analysis of GPS Software Receiver. *Second International Conference on Computer Modeling and Simulation*; 2010, p. 314-317.

[4] S. Abbasian, M. G. Petovello. Implementation of a Dual-Frequency GLONASS and GPS L1 /A Software Receiver. *The Journal of Navigation*; 2010; 63: 269﹣287.

[5] LI Shi, ZHAI Chuanrun, ZHAN Xingqun, WANG Boxiong. Implement of Real-Time GPS Ll Software Receiver. *Proceedings of 4th International Conference on Computer Science & Education*; 2009, p.1132-1137.

[6] Cai Bai-gen, ShangGuan Wei, Wang Jian , Liu Hui-chao. Design and Realization of Software Receiver Based on GPS Positioning Algorithm. *The 1st International Conference on Information Science and Engineering* ; 2009, p. 2030-2033.